Basic Stata
This is a brief guide to using the most basic operations in Stata. Stata also has an on-line tutorial. At the initial prompt type “tutorial.” In addition to the reference manuals and on-line help I would also recommend the following: Hamilton, Lawrence C. *Statistics with STATA* (Brooks/Cole).

1. Reading in Data
Stata starts in a default working directory. If you want to change this – for example if you want to work off an e drive, start by typing: cd “e:\”

a. Stata data sets
If you are fortunate enough to have your data in a Stata data set (*sample.dta*) on e drive, type:

```
use sample
```

b. Excel
If you are creating a dataset in Excel, make sure you only have one row of variable names. The variable names should be 32 characters or less, start with a letter and contain no spaces or special characters. Missing data should be coded either as blank or as a number (e.g. -999) and not as a period. The format of the data should be such that there are no commas. Save the data as type “CSV” (comma separated values). Read the file into Stata using the `insheet` command:

```
insheet using sample.csv
```
Now the data is in memory. To convert to a Stata dataset type: save sample
Stata will automatically append .dta to indicate that it is a Stata dataset.

c. Other formats
Data may be in an ASCII file with data separated by blanks or tabs and no variable names. Missing values may be indicated by a period or by a number. Use the `infile` command and give names to the variables as you read in the dataset: `infile spend income wealth using sample.raw`

For nonnumeric variables, use the command `str#` where # is a number equal to or greater than the number of characters in the variable: `infile income wealth str8 country using sample.raw`

You can use Stata’s data editor to create a dataset directly. There are also software programs that transform data from one format to another. Stat-transfer is one such (I have a copy you can use).
2. Creating Summary Statistics

For basic descriptive statistics of the whole dataset: \texttt{sum}

For descriptive statistics of only some variables: \texttt{sum income wealth}

For descriptive statistics on a specific group e.g. mean income for women aged 25-34:

\begin{verbatim}
sum income if female==1 & age >=25 & age <=34
\end{verbatim}

You can get more detailed information by typing: \texttt{sum income, d}

In addition to mean, standard deviation, min, max this will give you percentiles (1, 5, 10, 25, 50, 75, 90, 95 and 99), variance, skewness and kurtosis.

For categorical variables you can get frequency distributions (for one variable) or cross tabulations using the \texttt{tabulate} or \texttt{tab} command:

\begin{verbatim}
tab gender

\end{verbatim}

\begin{verbatim}
tab gender race

\end{verbatim}

Combine with the \texttt{sum} command to produce means of one variable for each value of another:

\begin{verbatim}
tab gender, sum (income)

tab gender race, sum (income)

\end{verbatim}

To look at observations, use the command: \texttt{list}

To look at one or two variables only: \texttt{list income wealth}

To look at individuals with an income less than $5000: \texttt{list if income>5000}

To look at a limited number of observations, for example the variables income and wealth for the 21st through 50th observations: \texttt{list income wealth in 21/50}

To look at all the variables in your dataset, including nonnumeric: \texttt{describe}
3. Running Regressions

a) OLS
Simple OLS regression where the dependent variable is spend and the independent variables are income and wealth:
\[
\text{reg spend income wealth}
\]
The regression command will produce coefficient estimates, standard errors, t-stats, p-values, 95% confidence intervals, F-test of all coefficients, R-squared, Adjusted R-squared, RSS, ESS, TSS.
To run a regression on a subset of data (e.g. just women, assuming there is a dummy variable female which is 1 for women):
\[
\text{reg spend income wealth if female==1}
\]
To calculate predicted values and residuals after a regression:
\[
\text{predict yhat}
\]
\[
\text{predict e, resid}
\]
Where yhat and e are the names of the predicted values and residuals respectively.

b) Dichotomous dependent variables
For probit and logit regressions where the dependent variable is 0 or 1 (e.g. the variable house might represent whether or not an individual buys a house) the commands are probit and logit:
\[
\text{probit house income wealth}
\]
The command will produce log likelihood, coefficient estimates, standard errors, z statistics, p-values, 95% confidence intervals, chi-squared test for all coefficients, pseudo R-squared.
Lots of initial stuff will appear on the screen – this is because Stata is going through an iterative process before it comes up with the final estimates.

4. Hypothesis Testing
To test whether there is a significant difference in mean income between two groups:
\[
\text{ttest income, by (female)}
\]
To test equality of two coefficients:
\[
\text{test income==wealth}
\]
To test a specific coefficient value:
\[
\text{test wealth==0.5}
\]
You might want F-tests of a certain subset of variables. For example, you may run a regression of spend on income and wealth and some dummy variables female white.
Test whether the dummy variables are jointly significant:
\[
\text{reg spend income wealth female white}
\]
\[
\text{test female white}
\]
This test works on the most recently run regression. The output is F-stat along with degrees of freedom and p-value.
4. Data Manipulation

a) Keeping and dropping variables
To keep only the income variable: keep income
To drop the income and wealth variables: drop income wealth
You may want to get rid of some part of your sample.
To keep individuals with only positive income: keep if income>0
To keep individuals with income $2000-$10000: keep if income>=2000 & income <=10000
To drop observations with income $1000 or more: drop if income >=1000
To restrict the sample to women: keep if female==1;
(Note the double equals sign tests for equality, whereas a single equals sign sets something equal to something else).

b) Transforming data
After you have loaded your data it may not be in exactly the form you want. Often, you want to create a new variable that is a function of an existing one. The syntax is: gen new = f(old)
For example:
  gen lninc = log (income)
  gen incsq = income *income
  gen incfem = income*female
To construct dummy variables:
  gen white = 1 if race == “Caucasian”
  replace white = 0 if race ==“AfricanAmerican”
(Since the original variable race is nonnumeric apostrophes are placed around it).

Another command that is frequently used for creating variables is egen. This is often used to create a variable that summarizes other variables:
  egen max_income = max (income)
If there are other transformations you want, the on-line help is good. Click on help and type gen or egen or use the commands: search gen or search egen
5. Using Two Data Sets

a) Merging datasets
You might want to combine different data sets together. To add a data set that has the same observations, but different variables, use the `merge` command. To add more observations of the same variables, use the `append` command. In either case, make sure that the data sets you are going to use are already Stata data sets. For both of these operations, it is assumed that one data set is already open and in memory and the other is stored on a drive.

Suppose one data set (data1) has information on household wealth and is in memory and another (data2) has household demographics and is on the e drive. If the datasets are organized the same way observation by observation (i.e. observation 1 is household 1 and so on in both data sets, then simply type:

```stata
merge using data2
```

Now data2.dta has been merged with the data in memory. A new variable gets created called `_merge`. It tells you the status of the merge: 1 means that the resulting observation occurred only in the data set that was in memory; 2 means that the observation only occurred in the data set that was on the drive; and 3 means that the observation has information from both. If households 1-20 were in my first data set (in memory) and 11-30 were in my second (on the drive), the merged data set would go from 1-30 with missing (though different) values for some variables. The variable `_merge` would be 1 for observations 1-10, 2 for observations 21-30 and 3 for observations 11-20.

Sometimes the two datasets are not organized as above, but have a variable identifying observations (like a household ID number). You can then you can merge on that variable.

```stata
merge hhid using data2
```

Now Stata has created a bigger data set by matching up observations by the variable `hhid`. The variable `_merge` is still created. If you need to merge multiple data sets, after each merge, you need to get rid of the variable `_merge` or the next merge won’t work. To do this type: `drop `_merge``

It is often advisable to sort data prior to a merge command, or for other reasons. You can sort by one or more specified variables:

```stata
sort year income
```

This sorts the data by year first and then by income within years.

b) Appending data sets
If you have two data sets with the same variables, but different observations, use the `append` command. This assumes there is one data set in memory and one on e drive.

```stata
append using data2
```

If there are variables in one data set that are not in the other, missing values will be generated.
6. Graphing
There are many things you can do to make fabulous-looking graphs. Here are the basics.

a) Scatter plots
To see the raw data relationship between income and wealth, type:

```
graph twoway scatter income wealth
```

b) Histograms
Histograms are good for giving an overview of the distribution of a variable. You can specify the number of bins you want:

```
hist income, bin(15)
```

c) Fitting regression lines
The following command will plot a partial regression plot for you if you have a multivariate regression. First, you need to estimate the relationship. To do this we use the `fit` command instead of `reg`. Then plot the graph using the command `avplot`:

```
fit spend wealth
avplot wealth
```
7. Writing .do files
You may want to get familiar with STATA by working interactively initially. However, ultimately the easiest way to use STATA is to create a .do file. What is a do file? It’s a program that performs commands in sequence and non-interactively. Create the .do file using the do-file editor in STATA and then run it by typing .do sample (where sample.do is the name of the file). Below is an example of a short .do file that assumes there is data in a STATA file called sample.dta. Comments are enclosed in /* */ so that STATA will not read them. This is a useful way to make notes to yourself about what you are doing in a program, particularly if the program gets long.

```stata
#delimit; /*sets the delimit, or end of sentence, to a semi-colon*/
log using sample.log, replace; /*opens up a log file to save your results, with replacement each time program is run*/
clear; /*makes sure nothing is in memory*/
use sample; /*loads the Stata dataset sample.dta*/
sum; /*calculates descriptive statistics*/
reg spend income wealth; /*runs a regression of spending on income and wealth*/
test income wealth; /*performs an F-test on income and wealth*/
test income==wealth; /*tests equality of two coefficients*/
test wealth==0.5; /*tests a specific coefficient value*/
gen lninc=log(income); /*creates a new variable*/
gen femed=female*ed /*creates an interaction variable*/
drop if famsize >=5; /*drops observations where family size is 5 or more*/
log close; /*closes the log file*/
```

All of your results will be saved in the log file. You can also directly print results in the Results screen using File – Print – Results.